CS232 Lab1
fa2020
streller

PROBLEM STATEMENT:

A review and extension of cs132: sort a file with 120 records. However, due to memory restrictions only 20 records may be placed into memory. You are to implement a "quasi" external sort

CODE/DIRECTIONS:

For the sake of simplicity, and without much loss of generality, we will say for this lab are records are just ints : thus sort a file with 120 ints where only 20 ints maybe placed into memory at any one time

general idea:
- break data into blocks, sort blocks into runs, merge runs

less general idea:
- Call inFile1 our source file (the one with the initial 120 records to be sorted.
- You will also need an inFile2, and 2 other files outFile1 and outFile2.
- Break the file into blocks of size 20: in this case there will be 6 blocks (120/20)
- Sort the blocks by
  - read a block, sort it, store in outFile1
  - read a block, sort it, store in outFile2
  - read a block, sort it, store in outFile1
  - (in other words, sort a block and alternately place in the files outFile1, outFile2)
  - By definition a run is a sorted block
  - Note that each file outFile1 and outFile2 will have half of the runs
- Merge the runs
  - Merge data from outFile1 and outFile2 to inFile1 and inFile2.
    - Merge the first run on outFile1 and the first run on outFile2, and store the result on inFile1:
    - Read two records in main memory, compare, store the smaller on inFile1
    - Read the next record from either outFile1 or outFile2 the file that had its record moved/stored to inFile1
    - Similarly merge the second run on outFile1 and the second run on outFile2, store the result on inFile2.
    - Merge the third run on outFile1 and the third run on outFile2, store the result on inFile1... etc
    - merging each run and storing the result alternatively on inFile1 and inFile2.
    - At the end
    - inFile1 and inFile2 will contain sorted runs twice the size of the previous runs on outFile1 and outFile2
  - Now merge data from inFile1 and inFile2 to outFile1 and outFile2.
    - Merge the first run on inFile1 and the first run on inFile2, and store the result on outFile1.
    - Merge the second run on inFile1 and the second run on inFile2, store the result on outFile2
    - Etc, merge and store alternatively on inFile1 and inFile2.
  - Repeat the process until only one run is obtained. This would be the sorted file

Must use above algorithm

MUST use algorithms from sort_algorithms.t:

the algorithms may not be modified. - but you may assume an overloaded < operator exists. Thus you may remove the pointer to the function cmp (obviously, the code will have to replace any reference to cmp, no puns intended, with <)

It *is permissible* to copy/paste any algorithms from sort_algorithms.t that you use to your own sort_alg.t

Declare a variable MAX_MEM_SIZE, of type size_t. Initialize to 20

If you are really paying attention to the directions, you will realize that 2 arrays of 20 cannot both be in memory at the same time – depending on how you implement the above algorithm and which sort you use, you may need to break into blocks of size 10

a sample file to be sorted is provided.

DELIVERABLES:

ONLY source files in a zip file named

cs232_lab1_YOURLASTNAME.zip

Submitted to blackboard CS232Lab1

The electronic submission is due at 8:00am 15 September 2020